

# WEB APPLICATION I

Written by Thapanapong Rukkanchanunt

# What is Web Application?

- **Web Application** คือการออกแบบซอฟต์แวร์ประยุกต์ที่ทำงานบนเว็บผ่านเครือข่ายอินเทอร์เน็ต
- ตัวอย่างของ **Web Application** ที่ได้รับความนิยมในปัจจุบันได้แก่
  - Bitdefender QuickScan (virus scan)
  - DeviantArt Muro (drawing tool)
  - Google Drive (office tools)
  - Ustream.tv (streaming tool)

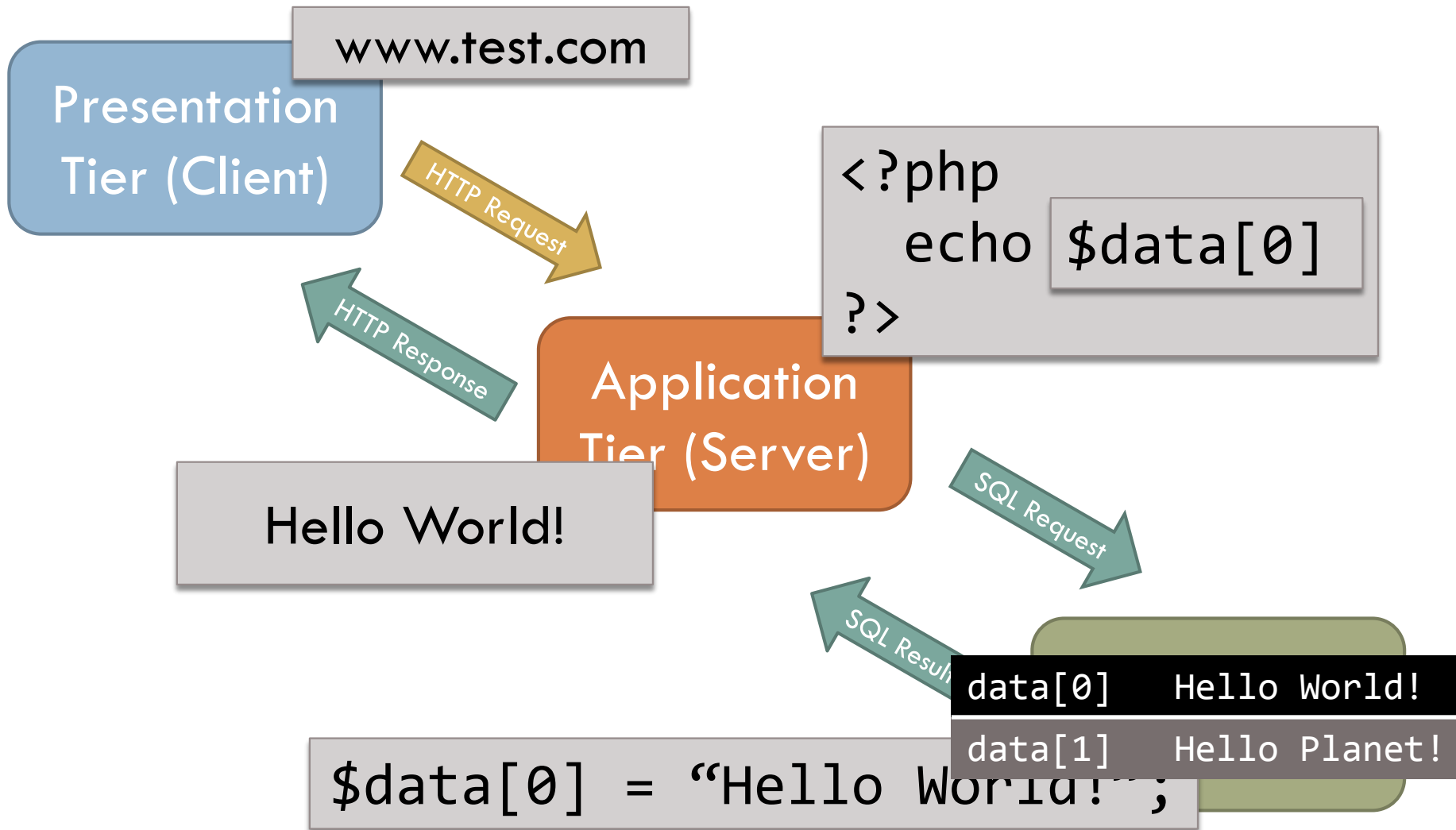
# How Web Application works

- โครงสร้างของ **Web Application** ที่ใช้อยู่ในปัจจุบันคือโครงสร้างแบบ **Client-Server Model**
- **Client-Server Model** เป็นโครงสร้างที่มีการแยกส่วนระหว่างฝั่งผู้ให้บริการหรือทรัพยากร (**Service Provider or Resource Provider**) ซึ่งเราจะเรียกว่าเป็น ฝั่ง **Server** และฝั่งผู้ขอใช้บริการหรือทรัพยากร (**Service Requester or Resource Requester**) นั้น ๆ ซึ่งเราจะเรียกว่าเป็น ฝั่ง **Client**
- ฝั่ง **Client** จะเป็นฝ่ายเริ่มต้นการสื่อสาร และ **Server** จะอยู่ในสถานะรอการเชื่อมต่อ

# Client-Server Architecture

- สถาปัตยกรรมแบบ **Three-tier** เป็นสถาปัตยกรรมที่มีการแยกการแสดงผล การประมวลผล และการจัดการข้อมูล ออกจากกัน
  - ▣ **Presentation Tier** – เป็น Tier ที่แสดงผลลัพธ์ของการประมวลผล ซึ่งจะเกิดขึ้นที่ฝั่ง **Client**
  - ▣ **Application Tier** – เป็น Tier ที่ประมวลผลข้อมูล และควบคุมการไหลของข้อมูลระหว่าง **Web Browser** ของ **Client** กับฐานข้อมูล
  - ▣ **Data Storage Tier** – เป็น Tier ที่จัดเก็บและดูแลข้อมูลทั้งหมด

# Three-tier Diagram



# Web Application Programming

- ในการเขียน **Web Application** นั้น เราสามารถเขียนหลายภาษาให้อยู่ในไฟล์เดียวกันได้ โดยแต่ละภาษานั้นจะถูกประมวลผลใน **Tier** ที่ต่างกัน ดังนั้นเราต้องเลือกภาษาให้ถูกต้อง
- **HTML, JavaScript, CSS** เป็นภาษาที่จะถูกประมวลผลที่ **Presentation Tier** ภาษาเหล่านี้จะควบคุมการแสดงผลในหน้าเว็บ
- **PHP** เป็นภาษาที่ถูกประมวลผลที่ **Application Tier** ภาษานี้จะควบคุมการไหลของข้อมูลระหว่าง **Client** และ **Data Storage** รวมถึงการคำนวณต่าง ๆ
- **MySQL** เป็นภาษาที่ถูกประมวลผลที่ **Data Storage Tier** ภาษานี้จะบันทึก แก้ไข ลบ และ เลือกข้อมูลเข้าและออกจากรฐานข้อมูล

# Web Application Programming [2]

- เราได้เรียนรู้การเขียน **Web Application** ด้วยภาษา **PHP** จากหลาย ๆ ปฏิบัติการที่ผ่านมา เราจะเรียกการกระทำนี้ว่า **Programming**
- **Programming** คือการถ่ายทอดขั้นตอนวิธี (**Algorithm**) ในรูปแบบสัญลักษณ์หรือภาษาที่คอมพิวเตอร์สามารถนำไปประมวลผลได้
- การที่เราจะได้ผลลัพธ์ที่ต้องการหรือไม่จะอยู่ที่ **Algorithm** เป็นหลัก ดังนั้นเราควรเรียนรู้ **Algorithm** ต่าง ๆ

# Basic Algorithm Rules

- ควรวางแผนก่อนการเขียนโปรแกรม
- คอมพิวเตอร์ทำงานตามคำสั่งจากบนลงล่าง ดังนั้นเราควรเรียงลำดับคำสั่งของเราให้ถูกต้อง (เหมือนการทำอาหาร ที่เราต้องทำตามลำดับ)
- เมื่อการใช้ตัวแปรใดในการคำนวณ ต้องแน่ใจก่อนว่าก่อนหน้าบรรทัดนี้ จะต้องมีการกำหนดค่าให้ตัวแปรนั้น ๆ ก่อนเสมอ
- ตรวจสอบการสะกดชื่อตัวแปร โดยเฉพาะการลืมเติม \$
- ห้ามลืม ; หลังคำสั่งทุกคำสั่ง



# Recommended Algorithm Outline

```
<?php
```

```
Declare functions
```

```
Define variables
```

```
Retrieve inputs
```

```
Compute results
```

```
Save results
```

```
Display results
```

```
?>
```

# Retrieving Inputs

- ในภาษา **PHP** เราสามารถรับข้อมูลเข้าจากผู้ใช้ได้หลายทาง เราจะค่อย ๆ เรียนรู้แต่ละทาง
- สำหรับวันนี้เราจะเรียนรู้เรื่องการรับข้อมูลด้วยวิธี **GET**
- การส่งข้อมูลจะทำได้ด้วยการกำหนดตัวแปรและค่าของตัวแปรไว้ใน **URL** เช่น **localhost:81 /index.php?n=5** เป็นการกำหนด **input** ที่มี **key** เป็น **n** และมี **value** เป็น **5**
- การส่งด้วยวิธี **GET** จะเก็บค่า **input** ไว้ในตัวแปร **Array** ที่ชื่อว่า **\$\_GET** ถ้าหากต้องการใช้ค่า **n** ที่มาจาก **URL** ข้างต้น เราต้องพิมพ์ **\$\_GET["n"]**

# Computing Results

- ในส่วนของการคำนวณผลลัพธ์ เราสามารถทำได้หลายวิธี โดยเราจะเรียนรู้โครงสร้างของการเขียนโปรแกรมทั้งหมด **3** แบบ
  - ▣ โครงสร้างแบบเรียงลำดับ (**Sequential Structure**)
  - ▣ โครงสร้างแบบมีเงื่อนไข (**Conditional Structure**)
  - ▣ โครงสร้างแบบวนซ้ำ (**Repetition Structure**)

# Sequential Structure

- โปรแกรมคำนวณพื้นที่ของรูปหลายเหลี่ยมด้านเท่า โดยรับข้อมูลเข้าเป็นจำนวนด้านและความยาวด้าน

```
$n = $_GET["n"];  
$s = $_GET["s"];
```

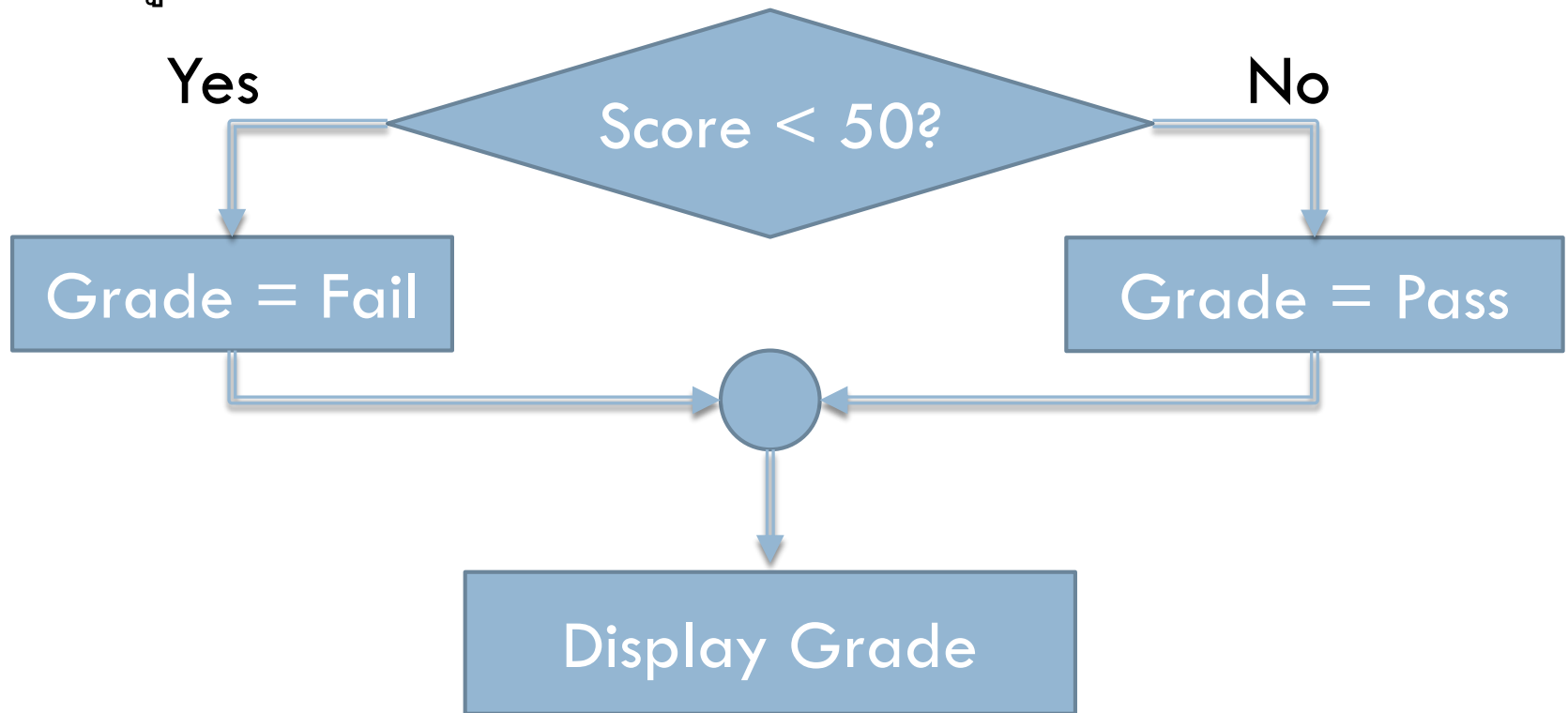
```
$area = 1/2*$n*$s*$s*1/tan(pi()/ $n);
```

```
echo "area is $area";
```

$$Area = \frac{1}{2} ns^2 \cot\left(\frac{\pi}{n}\right)$$

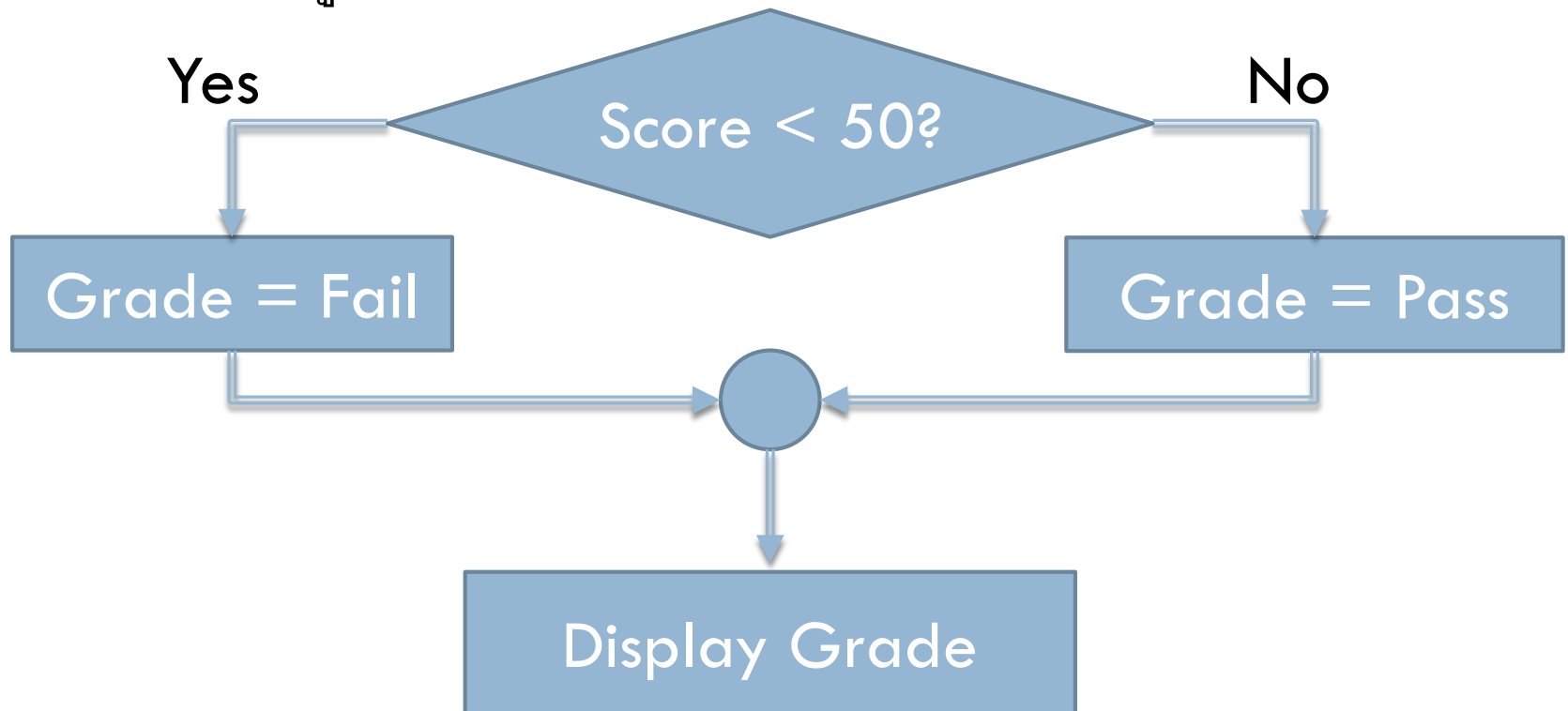
# Conditional Structure

- **Conditional Structure** เป็นโครงสร้างที่มีการทำงานแตกต่างกันออกไปขึ้นอยู่กับเงื่อนไขที่กำหนดให้



# Conditional Structure [2]

- จะสังเกตเห็นว่า **Grade** จะเป็นได้สองแบบคือ **Fail** หรือ **Pass** ขึ้นอยู่กับค่าที่เก็บอยู่ในตัวแปร **Score**



# Conditional Structure [3]

- ในภาษา PHP เราจะใช้คำว่า **if** ในการสร้างโครงสร้างแบบมีเงื่อนไข
- **Condition** จะต้องเป็นจริงหรือเท็จอย่างใดอย่างหนึ่งเท่านั้นต่อจนคำนวณ

Statements #1

```
if (condition) {  
    Statements #2  
} else {  
    Statements #3  
}  
Statements #4
```

# Conditional Structure Examples

```
<?php
    $score = $_GET["score"];
    if ($score < 50) {
        $grade = "fail";
    } else {
        $grade = "pass";
    }
    echo $grade;
?>
```



# Conditional Structure [4]

- ในกรณีที่เราไม่มีคำสั่งที่ต้องทำเมื่อ **condition** เป็นเท็จ เราสามารถลบส่วนที่เป็น **else** ออกไปได้เลย

```
<?php
```

```
    $x = $_GET["x"];
```

```
    if ($x < 0) {
```

```
        $x = -$x;
```

```
    }
```

```
    echo $x;
```

```
?>
```

# Conditional Structure Examples [2]

- จงเขียนโปรแกรมที่รับค่า  $x$  และ  $y$  แล้วแสดงผลค่าที่มากกว่า

# Conditional Structure Examples [3]

- จงเขียนโปรแกรมที่รับค่า  $x$  แล้วแสดงผลว่า  **$x$  is odd** ถ้า  $x$  เป็นจำนวนคี่ หรือ  **$x$  is even** ถ้า  $x$  เป็นจำนวนคู่

# Conditional Structure [5]

- ในกรณีที่เงื่อนไขเราต้องการเป็นไปได้มากกว่าสองทาง เราสามารถใช้ **elseif** แทน **else** ปกติได้

```
If (condition 1) {  
    Statement #1  
} elseif (condition 2) {  
    Statement #2  
} else {  
    Statement #3  
}
```

# Conditional Structure Examples [4]

```
<?php
if ($x > 0) {
    echo "x is positive";
} else if ($x == 0) {
    echo "x is zero";
} else {
    echo "x is negative";
}
?>
```

# Conditional Structure Examples [5]

```
<?php
if ($bmi > 0) {
    echo "Underweight";
} else if ($bmi > 18.5) {
    echo "Normal";
} else {
    echo "Overweight";
}
?>
```

# Conditional Structure Examples [6]

- จงเขียนโปรแกรมตัดเกรด โดยใช้เกณฑ์ดังนี้

Score	Grade
< 50	F
51 – 70	C
71 – 80	B
> 80	A

# References

---

- Gosselin, Don (2010) JavaScript: The Web Technologies Series, 5th Edition isbn: 0538748877